ERRATA SHEET

Date: **May 16, 2003**
Document Release: **Version 1.0**
Device Affected: **P89C669**

This errata sheet describes both the functional deviations and any deviations from the electrical specifications known at the release date of this document.

2003 May 16

Philips
Semiconductors

PHILIPS

Part identification:

P89C669 microcontroller is available in 44 pin PLCC and 44 pin LQFP package. Part can be identified by having one of these markings on the top of the package:

P89C669FA

P89C669BBD

Functional Deviations of P89C669

CORE.1: Code executed above 64 kB hit by interrupt when EIFM=0 ends out of application

Introduction: Interrupt vectors are located in microcontroller’s memory range 00:0003h to 00:0053h. Regardless of the current value of PC and EIFM, when interrupt occurs, PC is loaded with proper address from the above range. EIFM=1 forces microcontroller to push 3 byte long return address onto stack when interrupt is serviced, which is necessary in case interrupted instruction is located above 64 kB (systems with EAM=1). Having EIFM=0 and processing interrupt will result in pushing return address represented with 2 bytes only, and consequently polling 2 bytes only from the stack.

Problem: If a code located at address xy:abcd above 64 kB (i.e. xy<>00 and EAM=1) is interrupted in a system with EIFM=0, PC will not be loaded with content from page 0 (00:0003 to 00:0053) but with the content from page xy (range: xy:0003 to xy:0053), which is the wrong memory segment in this case.

Workaround: If EAM=1, EIFM must be set to EIFM=1 so that microcontroller can process interrupts correctly. In case EAM=1 and EIFM=0, at the bottom of every 64 kB page original interrupt vectors (located in range 00:0003 to 00:0053), should be copied. Even in this case, code executed above 64 kB (EAM=1) being interrupted in a system with EIFM=0, will result with unpredicted behaviour when RETI instruction is executed.

CORE.2: Access to off-chip XDATA with MOVX @DPTR instruction

Introduction: MOVX @DPTR instruction is going to access off-chip located data byte if content of DPTR is greater then the address of the last available byte of on-chip XDATA memory space (AUXR.1=0).

Problem: If instruction MOVX @DPTR type is located at 1:upper7bits:lower16bits and an address of location pointed at via DPTR is off-chip, microcontroller will access byte at 0:upper7bits:(DPTR) instead of byte at 0:00:(DPTR).

Workaround: Instead of using MOVX @DPTR, EMOV @PR0(1) instruction with R3(R7)=0 can be used in systems designed with 23 bits wide external address interface. Code written in C language should use pointers to far data in this case rather than pointers to XDATA.

CORE.3: MOVX @EPTR instruction produces unwanted activity on ALE and PSEN pins

Introduction: While MOV @DPTR can reach only 64 kB of external data, instruction MOVX @EPTR can access up to 8 MB - 64 kB of external data.

Problem: When MOVX @EPTR is used for external memory access in a system with ALE active all the time (AUXR.0=0), there is additional activity on PSEN pin after the read or write pulse. In a system where ALE is active only when access to external memory is performed (AUXR.0=1), there is extra activity on ALE and PSEN pins after the read or write pulse.

Workaround: None.

PCA.1: Toggle function in High-Speed Output Mode problems

Introduction: When selected, toggle option will enable square pulses to appear on selected PCA channel’s pin, if that channel is configured in High-Speed Output Mode.

Problem: Once pin dedicated to PCA channel (configured to operate in high-speed output mode), becomes controlled with toggle function, it can not be used as general purpose output digital pin from the rest of the application.

Workaround: General purpose digital output on this pin can be achieved if toggle function is first turned-off, then proper output level is set with write to port SFR, and toggle function is turned-on at the end again.

PCA.2: PCA Watchdog timer problems

Introduction: PCA Watchdog timer is used in order to prevent problems with application’s execution.

Problem: The PCA Watchdog timer function may not function properly at 24 MHz f_{osc} when PCA Count Pulse selection is set to “internal clock, $f_{osc}/2$“.

Workaround: None.

TIMER2.1: Manual setting of TF2 flag when Timer2 operates in Baud Rate Generator Mode fails

Introduction: TF2 can be used as “software interrupt“ in a system where Timer2 is configured to operate in Baud Rate Generator Mode, since with this setup Timer2 will not set TF2 when overflow happens, and therefore TF2 can be set by the rest of the code in order to initiate Timer2 interrupt service routine.

Problem: Manual setting of TF2 to TF2=1 in system where Timer2 is configured to operate in Baud Rate Generator Mode, will not trigger execution of Timer2 interrupt service routine.

Workaround: None.

TIMER2.2: Timer2 can not be used in Auto-reload Mode when Baud Rate Generator is used as clock source for UART0

Introduction: If Baud RateGenerator is selected to be source for receiving/transmitting clock for UART0, Timer2 can be configured to operate in any of non-Baud Rate Generator modes.

Problem: If Baud RateGenerator is selected to be source for receiving/transmitting clock for UART0, TF2=1 will not trigger execution of Timer2 interrupt service routine.

Workaround: None.

TIMER2.3: Timer2 is using clock in Programmable Clock-out mode different than the one specified

Introduction: Timer2 can be programmed to operate in Clock-out Mode, when 50% cycle clock with frequency derived from external oscillator is outputted on P1.0.

Problem: Instead of directly using external oscillator clock, Timer2 uses external oscillator clock divided by 6 to output 50% cycle clock on P1.0.

Workaround: None.

UART.1: RB8 and SBUF change in some configurations

Introduction: Bit SM2 enables multiprocessor communication. In order to wake up desired node in the network, 9 bit data communication must be established, using RB8.

Problem: Contents of RB8 and SBUF change when they should not if SM2=1 in modes 2 and 3.

Workaround: None.

UART.2: Double buffering in UART communication is not working

Introduction: In order to increase data output over UART using back-to-back transmission, double buffering can be selected.

Problem: Double buffering is not operational.

Workaround: None.

UART.3: MODE0 receive data sampled one clock later

Introduction: MODE0 is mode in which UART transmits/receives data and clock over its Tx/Rx lines.

Problem: UART MODE0 receive data is sampled one clock later than standard 80C51 UART does.

Workaround: None.

UART1.1: RxD1 pin is an open drain

Introduction: UART1 is functionally identical to UART0 and is used for asynchronous serial communication.

Problem: RxD1 pin is an open drain configuration.

Workaround: A resistor must be used if this pin is to be used as an output.

PORT4.1: PORT4 pins can not be used as general purpose I/O but UART1 only

Introduction: PORT4 consists of 2 pins that can be used either as general purpose I/Os or UART1 dedicated.

Problem: PORT4 pins can be used only as part of UART1.

Workaround: None.

ONCE.1: Problems with entering ONCE mode

Introduction: The ONCE (“On-Circuit Emulation”) Mode facilitates testing and debugging of systems without the device having to be removed from the circuit.

Problem: RxD1, TxD1 and ALE pins will not go into ONCE mode.

Workaround: None.